Naval Research Laboratory
Washington, DC 20375-5320

NRL/MR/6410--97-7937

The Stability and Multiplicity of the Monotonic Lagrangian Grid

ROBERT S. SINKOVITS
ELAINE S. ORAN
JAY P. BORIS

Lab for Computational Physics & Fluid Dynamics

May 28, 1997

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave Blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	May 28, 1997	

4. TITLE AND SUBTITLE

The Stability and Multiplicity of the Monotonic Lagrangian Grid

5. FUNDING NUMBERS

64-5666-0-7
62702E

6. AUTHOR(S)

Robert S. Sinkovits, Elaine S. Oran, and Jay P. Boris

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/MR/6410--97-7937

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington, VA 22203-1714

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

The Monotonic Lagrangian Grid (MLG) is a data structure in which nodes are ordered in a monotonic way such that those nodes which are close in physical space also have nearby indices in the data structure arrays. An MLG ordering for a given system of nodes, as defined by the monotonicity constraints, is not unique. For all but the smallest systems, the number of allowed orderings is extremely large with many of the possible MLG's so badly structured that they lead to poor results when used in physical calculations. A well-structured MLG ordering is one that corresponds well to the physical ordering of the system. This paper shows that the majority of the MLG's for a given set of node locations are poorly structured, but that the small fraction which are well-structured tend to be extremely stable against perturbations of the node positions. It is this extreme stability of the well-structured MLG's that is responsible for both the utility of this approach in particle-based simulations and the success of stochastic grid regularization, a technique for restructuring from a poorly structured to a well-structured MLG. The high probability of encountering a well-structured MLG when the node dynamics is complex, even without stochastic grid regularization, is a result of this relative stability.

14. SUBJECT TERMS

Monotonic Lagrangian Grid (MLG)
Relative stability
Nodes

15. NUMBER OF PAGES

28

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std 239-18
298-102

CONTENTS

THE STABILITY AND MULTIPLICITY OF THE MONOTONIC LAGRANGIAN GRID

I. INTRODUCTION

Particle-based simulations have become progressively more important over the last few decades owing to the tremendous advances in computer technology. Many of these techniques, including molecular dynamics (MD) [1], smooth particle hydrodynamics (SPH) [2], and Direct Simulation Monte Carlo (DSMC) [3], require that interactions or collisions be considered only between particles which are close in physical space. The local nature of the interactions makes it possible to drastically reduce the computational cost required in these simulations. For a system of particles interacting through pairwise forces, the computational cost for a simulation, in which forces are calculated between all pairs of particles, scales as $O(N^2)$. If the range of the interaction falls off quickly, the cost can be reduced to scale as $O(N)$. Research, primarily in the area of MD, has led to the development of a number of excellent algorithms for identifying and tracking neighboring particles. Many of these algorithms are based on the use of linked lists, neighbor lists, and trees.

Recently, alternative particle tracking techniques based on the Monotonic Lagrangian Grid (MLG) [4] have been developed. The MLG is a multidimensional data structure in which objects which are close in physical space are assigned sets of indices which are close in logical or integer space. An advantage of the MLG vis-a-vis most other approaches is related to the relative locality of all computations and thus a relative computational advantage on distributed memory parallel computers.

The locations of the objects within the MLG data structure are determined by a set of monotonicity conditions on the physical coordinates. As an example, consider a system of particles in two dimensions. The coordinates, velocities, and other attributes of the particles can be stored in a set of two-dimensional arrays $x(N_x, N_y)$,

Manuscript approved February 29, 1997.

$y(N_x, N_y)$, $v_x(N_x, N_y)$, etc., where $N = N_x \times N_y$ is the total number of particles. The particles are said to be in MLG order when the coordinates satisfy the following sets of inequalities

$$x(i,j) \leq x(i+1,j) \qquad 1 \leq i \leq N_x - 1 \qquad (all\ j), \tag{1a}$$

$$y(i,j) \leq y(i,j+1) \qquad 1 \leq j \leq N_y - 1 \qquad (all\ i). \tag{1b}$$

For a system of particles which are in MLG order, particles which are close in physical space have indices (i,j) which are close in logical or integer space. Identification of the neighbors of a given particle is carried out by searching a specified region of the index space. The maximum index offset N_c required to identify all pairs of interacting particles is determined by the density of the system and the cutoff distance R_c of the interaction. Although it is almost impossible to guarantee that a given choice for N_c will ensure that all interacting pairs of particles are found, previous work has shown that, for reasonably small values of N_c, the probability of missed interactions can be made vanishingly small [5]. A timing analysis of the MLG on various test problems shows that it performs well at identifying all pairs of interacting particles [6].

Until recently, the MLG has been used primarily in MD simulations [7–10]. Applications have included the study of shock-induced detonations in energetic materials [7], $(N_2)_2$ dimer formation [8], and shock-defect interaction in Lennard-Jones solids [9]. The MLG has also been used in related problems in battle management [11].

Although the MLG was originally developed for particle tracking in MD, the greatest utility may lie in DSMC applications [13–16]. DSMC is a particle-based technique used to model high Knudsen number flow, $(Kn = \lambda/L > 0.1)$, where λ is the mean free path of the gas molecules and L is the smallest characteristic physical dimension of the system. Under these conditions, solutions of the Navier-Stokes

equations give physically incorrect results. The DSMC technique is not based on the Boltzmann equation, but can be shown to give results that are consistent with it.

In a DSMC simulation, particle motions are modeled deterministically, whereas binary collisions between particles are treated statistically. In the standard DSMC approach, physical space is divided into cells and collision pairs are chosen randomly from particles belonging to the same cell. Macroscopic properties are calculated in each cell by taking the appropriate averages of particle quantities within the cells. There are a number of disadvantages to this Eulerian cell-based approach, particularly when there are large spatial or temporal variations in the density of the system or complex geometries are involved. Since DSMC is inherently a statistical technique, large errors can be associated with cells which contain small numbers of particles. One of the difficulties involves partitioning space in such a way so as to have a relatively constant number of particles in each cell. A second difficulty, particularly in the case of non-orthogonal or unstructured grids, is the assignment of the particles to the proper cells. In an MLG-based DSMC code, the locations of the particles can be used to define the physical grid. The DSMC-MLG approach provides both automatic grid refinement for transient flows and ensures that each physical cell contains exactly the same number of particles. This last condition is not strictly necessary in a DSMC calculation, but is advantageous since it leads to increased accuracy of the solution, simplified coding, and more efficient parallelization [16].

Since the MLG was first introduced, it was known that the MLG ordering for a given spatial distribution of objects is not unique. This is consistent with Eqs. (1) since the number of coordinate inequalities used to define the MLG order is less than the number of degrees of freedom of the system. Figure 1 shows that there can be significant variation in the quality of the MLG orderings, where the quality is a measure of how well the MLG ordering corresponds to the physical ordering of the system. For

both MD and DSMC simulations, it is advantageous to work with high-quality MLG orderings. In MD simulations this follows from the fact that the computational cost associated with the evaluation of the short-range interactions is proportional to the volume of index space that must be searched. In DSMC calculations, the validity of the results diminishes when poorly-structures MLG orderings are used since random collision pairs may involve particles that are no longer close in physical space.

The variation in MLG quality for a single collection of node locations has led to the development of stochastic grid regularization (SGR) [12], an efficient technique for optimizing the structure of an MLG. A single iteration of SGR consists of: (1) Randomly displacing nodes in space, retaining information about the positions of the unperturbed nodes, (2) Exchanging data between adjacent perturbed nodes until monotonicity conditions are satisfied for the displaced node positions, and (3) Starting from the MLG ordering established in step (2) for the perturbed node positions, exchanging data between adjacent nodes until the monotonicity conditions are satisfied for the unperturbed node positions.

The perturbed node positions are obtained by independently choosing displacements along each coordinate axis from a uniform random distribution. The magnitude of the maximum node displacement is the critical parameter in SGR. The optimal choice for the maximum node displacement is approximately one-half the average internode separation [12]. Multiple iterations of the SGR in a single simulation timestep may further improve the grid structure. In addition to being straightforward and relatively easy to apply, one of the primary advantages of SGR is that it tends to restructure only the most poorly structured regions of the grid while leaving the majority of the well-structured regions relatively unchanged.

Previously we have shown that SGR is effective at restructuring an MLG [12], but it was not explained why SGR works so well. Critical to our explanation for the

success of SGR is the concept of MLG stability. The basic idea proposed in this paper is that SGR allows the system to access the more stable MLG's which also tend to be well-structured. Our definition of MLG stability is different from standard definitions of stability. In this context, we mean that an MLG is stable if the MLG inequalities are not violated for small displacements of the nodes.

Before addressing the questions of stability, we first attempt to make theoretical estimates of the number of allowed MLG orderings. In addition, for small systems of nodes, all possible MLG orderings are generated. These results are important not only because they demonstrate the richness of the MLG data structure, but because they provide us with an opportunity to study the properties of a complete distribution of MLG orderings. Next, several grid diagnostics are developed and the concept of grid stability is introduced. We show that poorly structured regions of the grids tend to be unstable and that the use of SGR leads to more stable grids. Finally, the grid diagnostics are applied to the complete sets of MLG orderings to show that not only are the well-structured grids more stable, but that the majority of the MLG's are both poorly structured and extremely unstable.

II. ESTIMATES OF MLG MULTIPLICITIES

In all but certain special cases, many MLG's may be found for a given set of node locations. In this section, we establish upper bounds on the number of MLG orderings for a given set of nodes and explicitly generate all possible MLG orderings for small systems in two dimensions. For simplicity and clarity, all derivations and numerical experiments described in this section are for the case of a two-dimensional MLG, although they can be generalized to higher-dimensional systems.

A theoretical estimate of the upper limit on the number of allowed MLG orderings, n_{max}, can be derived for a set of $N = N_x \times N_y$ nodes. First, assume that the

coordinates of the nodes are not degenerate, *i.e.*, no two nodes have the same x or y coordinates. This assumption is justified in applications such as molecular dynamics where the spatial coordinates take on a continuous range of values and the probability of two particles having identical x or y coordinates is negligible. If we assume there are no restrictions on the orderings such as those in Eqs. (1), there are $N!$ ways of sorting N nodes into an $N_x \times N_y$ data structure. The nodes can be sorted into the data structure by first selecting N_x nodes out of N to form the bottom row, then selecting N_x nodes out of $N - N_x$ to form the second row and so on until all the rows have been constructed. There are $N_x!$ allowed permutations of the nodes in each row, resulting in $N!$ ways of sorting the nodes into the data structure. If the MLG constraints on the x-coordinates are imposed, only a single permutation of the nodes in each row is allowed, reducing the upper limit on possible MLG orderings to $n_{max} = N!/(N_x!)^{N_y}$. This limit can be further reduced by imposing the MLG inequalities on the y-coordinates. There is only one ordering of the N_y rows such that the coordinate inequalities on the first column of nodes are satisfied. This reduces the upper limit on the number of MLG's to $n_{max} = (N!/(N_x!)^{N_y} N_y!)$. This line of reasoning is also valid if the nodes are first sorted into columns and then the coordinate inequalities on the first row of nodes is imposed, with the roles of N_x and N_y exchanged in the final expression. Since we are deriving upper bounds to the number of MLG orderings, the expression which yields the smaller number of MLG's should be taken as the upper limit. At this point, the orderings of nodes are analogous to a tangled mop head in which one end of each strand is fixed and all the strands are allowed to weave and cross as long as they do not bend back on themselves. Unfortunately, after the monotonicity conditions on the rows and a single column are satisfied, it is impossible to continue the combinatorial arguments along these lines without taking into consideration the actual placements of the nodes. This result is

not surprising since for systems as small as four nodes, the number of allowed 2×2 MLG orderings is not unique, but depends on the actual spatial arrangement of the nodes, as shown in Figure 2.

Attempts were made to generate all of the allowed MLG orderings for small systems of nodes. The purpose of these calculations was two-fold. The first was to determine the number of MLG's for typical sets of nodes that might be encountered in a molecular dynamics or DSMC calculation where the nodes are spread relatively evenly in physical space. The second purpose is perhaps more important. Once we have a full set of MLG orderings, it is possible to determine the range and distribution of MLG properties. A brute force approach in which all $N!$ permutations of the nodes are tested against the MLG coordinate inequalities becomes impractical for systems of size larger than 3×3. A more efficient algorithm used on systems up to size 5×5 is described in detail in the appendix.

Table 1 shows the range in the number of allowed MLG's for typical node arrangements for 2×2, 3×3, 4×4 and 5×5 systems. Although the numbers of allowed MLG orderings are very small compared to the estimates of the upper-bound n_{max} derived above, there are still a very large number of possible orderings for even small systems. The largest system (5×5) for which we can explicitly generate all allowed orderings is at least an order of magnitude smaller than a minimum-size system that would be used in an MD simulation of a gas or liquid. Still, for a system of this size, there are typically more than 10000 allowed MLG orderings. The number of MLG orderings increases very rapidly with system size.

For all but the smallest systems, exact enumeration of all MLG's becomes impractical. A Monte Carlo approach in which node orderings are chosen at random and then tested against coordinate inequalities is also not a viable alternative for estimating the number of MLG orderings since the fraction of node permutations which

satisfy the MLG inequalities is very small. In another approach, numerical experiments were carried out in which the nodes were placed in a random order and then restored to MLG order by swapping data between adjacent nodes until the monotonicity conditions are satisfied. This technique can only be used to give a lower bound on the number of orderings since it is impossible to be certain that all MLG's are found. In fact, it is likely that the majority of the poorly structured MLG's are not encountered. Such a calculation on a 6×6 system indicates that a minimum of 75,000 MLG's are allowed for a typical arrangement of nodes. For systems of sizes normally used in molecular dynamics simulations, the number of orderings is expected to be exceedingly large.

III. MLG DIAGNOSTICS AND STABILITY

Various statistical quantities can be used to give a good measure of the overall quality of an MLG. One useful set of parameters is the average link lengths between index neighbors in the MLG, where the individual links lengths are defined as

$$x_{link}(i,j) = |\mathbf{r}(i+1,j) - \mathbf{r}(i,j)|, \tag{2a}$$

$$y_{link}(i,j) = |\mathbf{r}(i,j+1) - \mathbf{r}(i,j)|. \tag{2b}$$

For a set of MLG's corresponding to the same set of nodes, the ones with the smaller average link lengths tend to be better structured. Another useful set of parameters is the averages of the normalized dot products of vectors joining near neighbors with the unit normals. The normalized dot products are defined in terms of the node positions and link lengths as

$$x_{dot}(i,j) = \frac{x(i+1,j) - x(i,j)}{x_{link}(i,j)}, \tag{3a}$$

$$y_{dot}(i,j) = \frac{y(i,j+1) - y(i,j)}{y_{link(i,j)}}. \tag{3b}$$

The average dot products are a measure of the directionality of the links joining near neighbors in index space. For a set of nodes arranged on a square lattice, or a higher-dimensional generalization of a square lattice, it is possible to find an optimal MLG in which all dot products are equal to one. For random distributions of nodes, the MLG's with the larger average dot products are better structured.

The grid diagnostics defined in Eqs. (2) and (3) can be averaged over all axes to give an overall measure of the quality of the MLG. However, because it is possible for an MLG to have diagnostic values indicative of good grid quality along one axis and not the others, it is useful to retain the quantities over each individual axis. This is illustrated in Figure 3 which displays the values of the grid diagnostics for the complete set of 5×5 MLG's corresponding to a single set of 25 nodes. The majority of the MLG's have values for the diagnostics that lie within a relatively small range, while a small fraction of the MLG's have either very good or very poor diagnostic values.

The average diagnostics defined above are useful measures of MLG quality and are relatively inexpensive to compute, but they may underestimate the influence of small locally distorted regions of the grid. This is important because it is the most poorly structured regions of the grid that determine the size of the maximum index offset required in a molecular dynamics simulation. The use of SGR has been shown to result in better-structured grids, but the improvements due to SGR are only weakly reflected in the average grid diagnostics. To overcome the weaknesses of the average grid diagnostics and to provide an explanation for the success of SGR, we have developed the concept of grid stability.

The locations of N nodes embedded in a d-dimensional space can be described by

a single point in an Nd-dimensional coordinate space. As the positions of the nodes are perturbed, the point describing the state of the system traces out a volume in the Nd-coordinate space. We can imagine ordering the nodes in a particular MLG and then perturbing the node positions. For certain displacements of the nodes, the MLG coordinate inequalities remain satisfied, while other displacements result in a violation of the coordinate inequalities. We define MLG stability to be proportional to the volume of coordinate space over which the system of nodes can be perturbed without violating the monotonicity conditions for the grid.

In practice, the calculation of the volume of coordinate space is not practical since it involves simultaneously perturbing each node over each of its degrees of freedom. As a lowest-order approximation, the volume of coordinate space can be estimated by calculating the volume of an Nd-dimensional box whose sides are defined by the maximum allowed displacements of each node over each degree of freedom. We denote this estimate of MLG stability as S_1 since it is calculated by perturbing a single coordinate at a time. Higher-order estimates S_n would be calculated from simultaneous displacements of n coordinates.

Regardless of the order of approximation used in calculating MLG stability, restrictions must be imposed on the displacements of the nodes so that the volume of coordinate space remains finite. Uniform translations of all nodes or breathing displacements in which all node coordinates are multiplied by a constant avoid violating the MLG coordinate inequalities, but lead to infinite contributions to the volume of coordinate space. The calculation of S_1 is considerably simpler since the only restriction that must be imposed is that edge nodes cannot be displaced from the remaining nodes. An example of a disallowed displacement would be to move node (N_x, j) in the positive x direction, where N_x defines the first dimension of the data structure and j is arbitrary. Taking into account the restrictions on displacements of edge nodes, the

definition for S_1 for a 2d system is

$$S_1 = \prod_{i=1,N_x-1;\text{all}j} (x(i+1,j) - x(i,j)) \prod_{\text{all}i;j=1,N_y-1} (y(i,j+1) - y(i,j)). \qquad (4)$$

The success of SGR can be explained in terms of the MLG stability. The regions of the grid that have the greatest effect in reducing the MLG stability are exactly those that are most poorly structured. At these locations, small displacements of the nodes can lead to violations of the MLG coordinate inequalities. In the well-structured regions of the grid, the nodes can be perturbed significantly before the monotonicity conditions are violated. This is illustrated in Figure 4. By perturbing the node positions, the poorly structured regions are given the opportunity to settle into a more stable MLG configuration. We have found that there is an optimal maximum node displacement of approximately one-half the internode separation. The existence of an optimum displacement can be explained by the fact that node displacements that are too small do not cause a restructuring of the locally unstable regions of the grid while displacements that are too large result in an unnecessary global restructuring of the grid.

Figure 5 shows the MLG stabilities as a function of the average link lengths and average dot products. In contrast to the average grid diagnostics which have values that are constrained to lie within a narrow bounds, variation in the MLG stability extends over approximately 14 orders of magnitude. Since the quantity S_1 is computed as a product over the projections of all the links, it is expected that the range in stability would increase with system size. The majority of the MLG's are clustered around a central value, while a small fraction lie in the tails of the distribution with very high or very low MLG stabilities.

Linear correlation coefficients have been calculated between $\log(S_1)$ and the average dot products and link lengths. The linear correlation coefficient between a pair

of variables x and y is given by

$$r(x,y) = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{(x_i - \overline{x})^2}\sqrt{(y_i - \overline{y})^2}}. \tag{5}$$

Although the values of the coefficients, $r(\log(S_1), \overline{dot}) = 0.845$ and $r(\log(S_1), \overline{link}) = -0.748$, indicate a relatively strong correlation, the MLG stability can vary over several orders of magnitude for a given value of average dot product or link length. For well-structured grids, the variation in stability at a fixed value of average grid diagnostic can still be large, but is narrower than for the poorly structured grids.

IV. CONCLUSIONS

Even for small systems of nodes, the number of MLG orderings can be extremely large and there can be significant variation in the number of MLG's depending on the spatial arrangements of the nodes. Extrapolating the results to systems of sizes typical in molecular dynamics simulations, the number of possible MLG's becomes, for all practical purposes, infinite. Given the expense associated with generating different MLG's and the large number of possible MLG orderings, an exhaustive search through all allowed orderings for the optimal MLG is impossible.

By examining complete sets of MLG's for small systems, we have found that the majority of the MLG's are low quality, as measured by the average link lengths and dot products. At first this result would seem to indicate that the MLG approach is not satisfactory for the identification of neighboring nodes. Fortunately, even without using grid optimization techniques such as SGR, the most poorly structured MLG's are very rarely encountered. In fact, it is quite difficult to find the worst MLG's even if the nodes are sorted from random order into MLG order. This is due to the strong correlation between grid quality and stability. The small number of MLG's that are well-structured are also extremely stable.

The grid perturbation that is part of SGR is implicitly included in MLG-based molecular dynamics simulations. The MLG data structure is updated periodically every timestep or every few timesteps during the simulation. The motion of the particles effectively perturbs the system, putting it slightly out of MLG order. Repeatedly sorting the particles into MLG order from previous MLG orderings leads to stable, well-structured grids.

Often MLG's are found that are generally well-structured, but which contain local regions that are very poorly structured. SGR optimizes the MLG by restructuring the locally distorted regions and driving the system in the direction of increased stability, while leaving the well-structured regions of the grid relatively unchanged. Improvements in the grids that are reflected only weakly in the average dot products and link lengths result in significant improvements in the MLG stability, as measured by the lowest-order estimate S_1.

ACKNOWLEDGMENTS

This work was supported by the Office of Naval Research and the Applied and Computational Mathematics Program at DARPA. RSS thanks the Office of Naval Technology for support as a ONT Postdoctoral Fellow. The authors also thank David Fyfe and Ronald Kolbe for helpful discussions and insights regarding this work.

REFERENCES

[1] R.W. Hockney and J.W. Eastwood, *Computer Simulation Using Particles*, Institute of Physics Publishing Ltd, Bristol (1988).

[2] J.J. Monaghan and R.A. Gingold, *Shock simulation by the particle method SPH*, J. Comput. Phys., 52 (1983), pp. 374-389.

[3] G.A. Bird, *Molecular gas dynamics and the direct simulation of gas flows*, Clarendon Press, Oxford (1994).

[4] J.P. Boris, *A vectorized "near neighbors" algorithm of order N using a Monotonic Logical Grid*, J. Comput. Phys., 66 (1986), pp. 1-20.

[5] S.G. Lambrakos and J.P. Boris, *Geometric properties of the Monotonic Lagrangian Grid algorithm for near-neighbor calculations*, J. Comput. Phys., 73 (1986), pp. 183-202.

[6] J.M. Picone, S.G. Lambrakos and J.P. Boris, *Timing analysis of the Monotonic Logical Grid for many-body dynamics*, SIAM J. Sci. Stat. Comput., 11 (1990), pp. 368-388.

[7] S.G. Lambrakos, M. Peyrard, E.S. Oran and J.P. Boris, *Molecular-dynamics simulations of shock-induced detonations in solids*, Phys. Rev. B, 39 (1989), pp. 993-1005.

[8] S.G. Lambrakos, J.P. Boris, R.H. Guirguis, M. Page and E.S. Oran,*Molecular dynamics simulation of* $(N_2)_2$ *formation using the Monotonic Lagrangian Grid*, J. Chem. Phys., 90 (1989), pp. 4473-4481.

[9] L. Phillips, R.S. Sinkovits, E.S. Oran and J.P. Boris, *The interaction of shocks and defects in Lennard-Jones crystals*, J. Phys.: Condens. Matter, 5 (1993), pp. 6357-6376.

[10] J.H. Dunn and S.G. Lambrakos, *Calculating complex interactions in molecular dynamics simulations employing Lagrangian particle tracking schemes*, J. Comput. Phys., 111 (1994), pp. 15-23.

[11] R.L. Kolbe, J.P. Boris and J.M. Picone, *Battle engagement area simulator/tracker*, NRL Memorandum Report 6705 (1990).

[12] R.S. Sinkovits, J. Boris and E.S. Oran, *A technique for regularizing the structure of a Monotonic Lagrangian Grid*, J. Comput. Phys., 108 (1993), pp. 368-372

[13] B.Z. Cybyk, *Combining the Monotonic Lagrangian Grid with Direct Simulation Monte Carlo; a new approach for low-density flows*, Ph. D. Thesis, University of Maryland (1994).

[14] B.Z. Cybyk, E.S. Oran, J.P. Boris, and J.D. Anderson, *Combining the Monotonic Lagrangian Grid with a Direct Simulation Monte Carlo Model*, submitted to J. Comput. Phys. (1994).

[15] B.Z. Cybyk, E.S. Oran, J.P. Boris, and J.D. Anderson, *Application of the Monotonic Lagrangian Grid in Direct Simulation Monte Carlo methods*, AIAA Paper 94-0345 (1994).

[16] C.K. Oh, R.S. Sinkovits, B.Z. Cybyk, E.S. Oran, and J.P. Boris, *Massive parallelization of DSMC combined with the Monotonic Lagrangian Grid*, AIAA Paper 95-0409 (1995).

APPENDIX: EXACT ENUMERATION OF MLG'S

For systems up to size 5×5, it was still possible to explicitly generate all allowed MLG orderings. The search through all 25! possible node orderings was avoided by using the algorithm described below. Approximately 20 minutes of CPU time was required on a SUN-4 to generate the complete set of 5×5 MLG orderings for a typical system of 25 nodes.

The nodes are first sorted in order of increasing x-coordinate. In addition to the x- and y-coordinate values, each node is assigned an index which gives the rank of the y-coordinate. Just by ordering the nodes in ascending x, the problem is reduced from a search over $N!$ to $N!/(N_x!)^{N_y}$ orderings. The outer loop of the program is over the $N!/(N - N_x)!N_x!$ allowed choices for the bottom row. The second row is then chosen from the remaining $(N - N_x)!/(N - 2N_x)!N_x!$ possibilities. Proceeding in this fashion, the search for all MLG orderings would require testing $N!/(N_x!)^{N_y}$ permutations. Computational savings can be realized by immediately testing the y-coordinate inequalities of row-2 against row-1 before proceeding to choose the next row. In this way, calculations for a large number of node permutations can be avoided by recognizing violations of the MLG inequalities before the entire grid is constructed. Testing adjacent rows for satisfaction of the y-coordinate inequalities is performed after the choice of each new row.

Additional computational savings can be realized by exploiting the fact that particular nodes must lie within certain rows. For example, the node with the minimum y-coordinate must lie within the bottom row of the MLG. This can be shown by the following argument. Allow the node with the minimum y-coordinate to lie in the second row. For the MLG inequalities to be satisfied, a node would have to be found with a lower y-coordinate. Since such a node does not exist, the node with the

minimum y-coordinate must lie in the bottom row. Similarly, it can be shown that the node with the second lowest y-coordinate must lie within the bottom two rows. This argument can be extended to the $(N_y - 1)$ nodes with the lowest y-coordinates. In a completely analogous fashion, the node with the maximum y-coordinate must lie within the top row, etc. These conditions can be tested immediately upon construction of the rows to quickly recognize node configurations that will not satisfy the MLG inequalities. The results derived are completely general for any spatial arrangement of nodes. For particular node arrangements, more constraints on the placements of nodes within particular rows could be derived, thereby making the extension to slightly larger systems computationally feasible.

TABLE 1. Number of MLG orderings for systems of various sizes. Range in values was obtained by explicitly generating all possible MLG orderings for different spatial arrangements of nodes.

No. of nodes	MLG shape	No. of MLGs
4	2×2	1-2
9	3×3	3-12
16	4×4	91-405
25	5×5	10130-97799

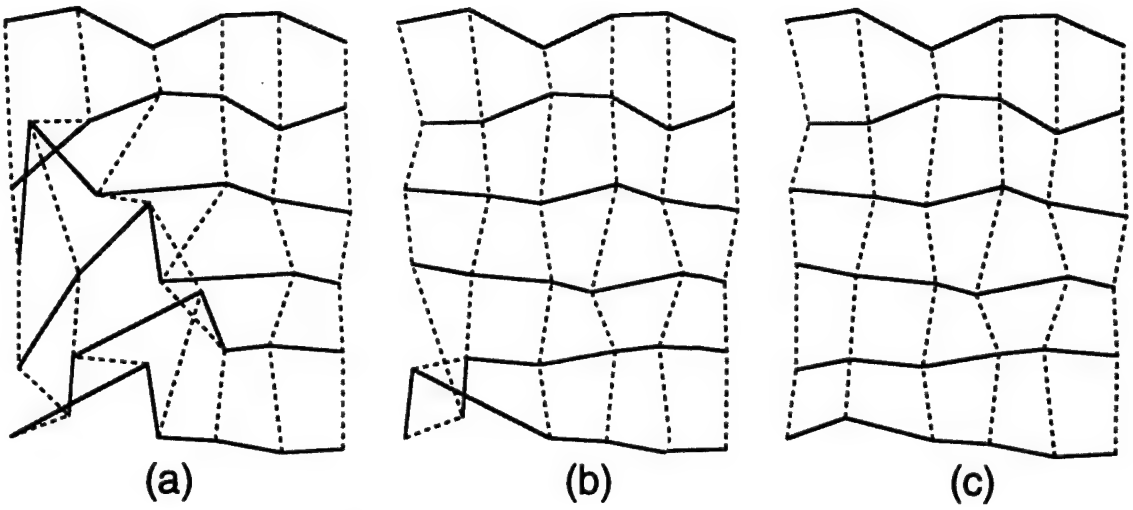

Fig. 1 — Example of three different Monotonic Lagrangian Grids based on the same set of node locations. Solid and dashed lines represent x and y links respectively. Grid (a) was obtained by sorting nodes into MLG order from a random order. Grids (b) and (c) are derived from grid (a) by 1 and 2 iterations, respectively, of SGR.

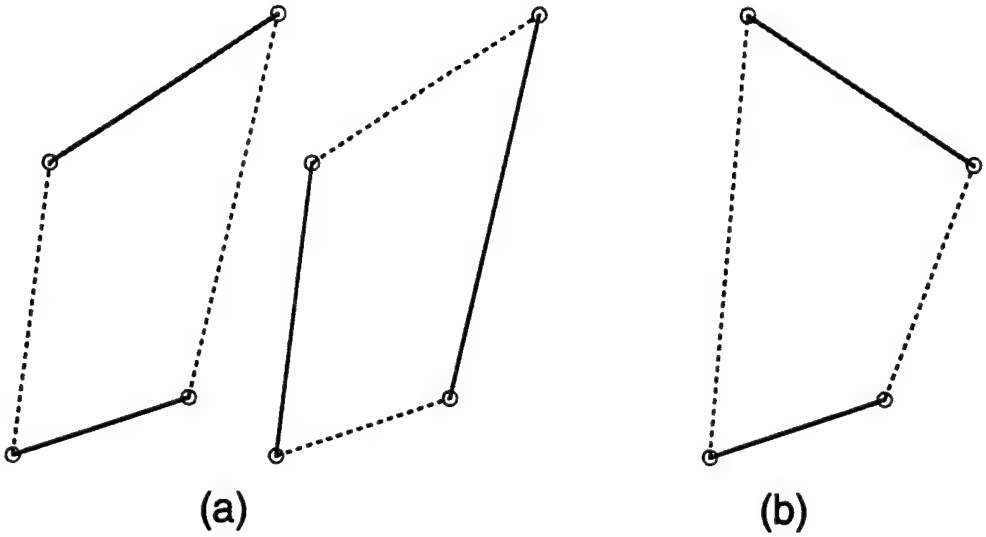

Fig. 2 — Allowed 2 x 2 MLG orderings for four nodes in two dimensions. Solid and dashed lines represent x and y links respectively. Configuration (a) has two allowed MLG's while configuration (b) has one possible MLG.

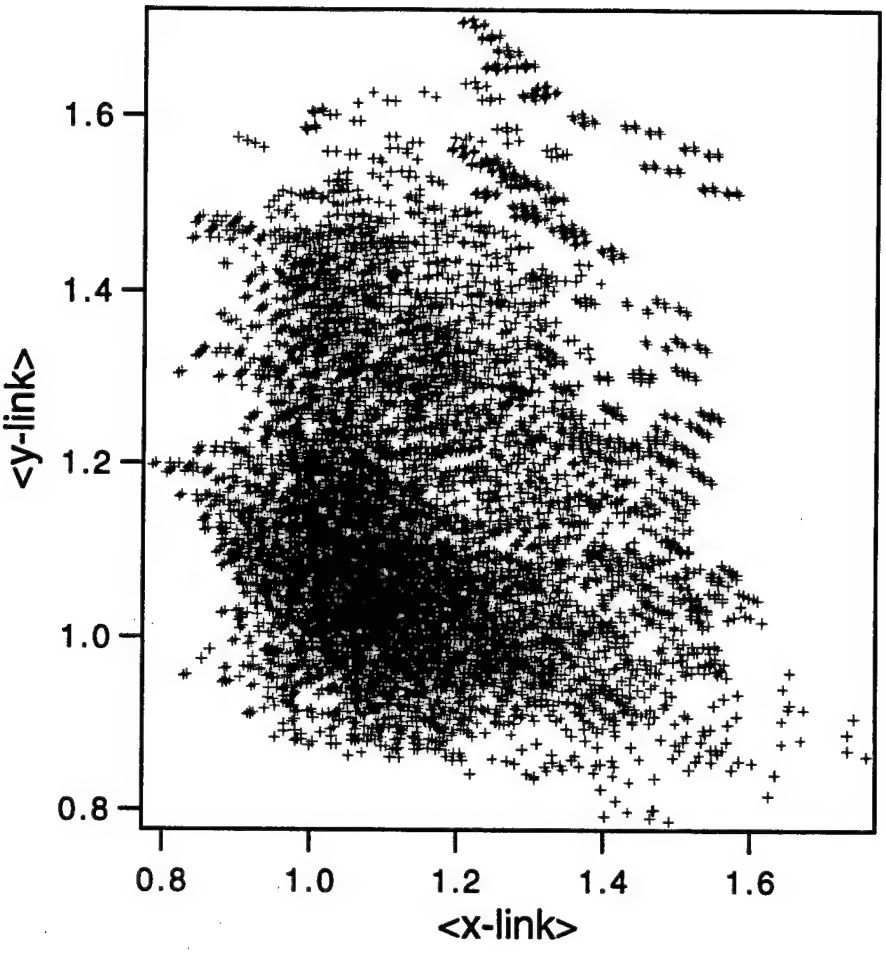

Fig. 3(a) — Average x and y link lengths for all allowed 5×5 MLG's for a set of 25 nodes.

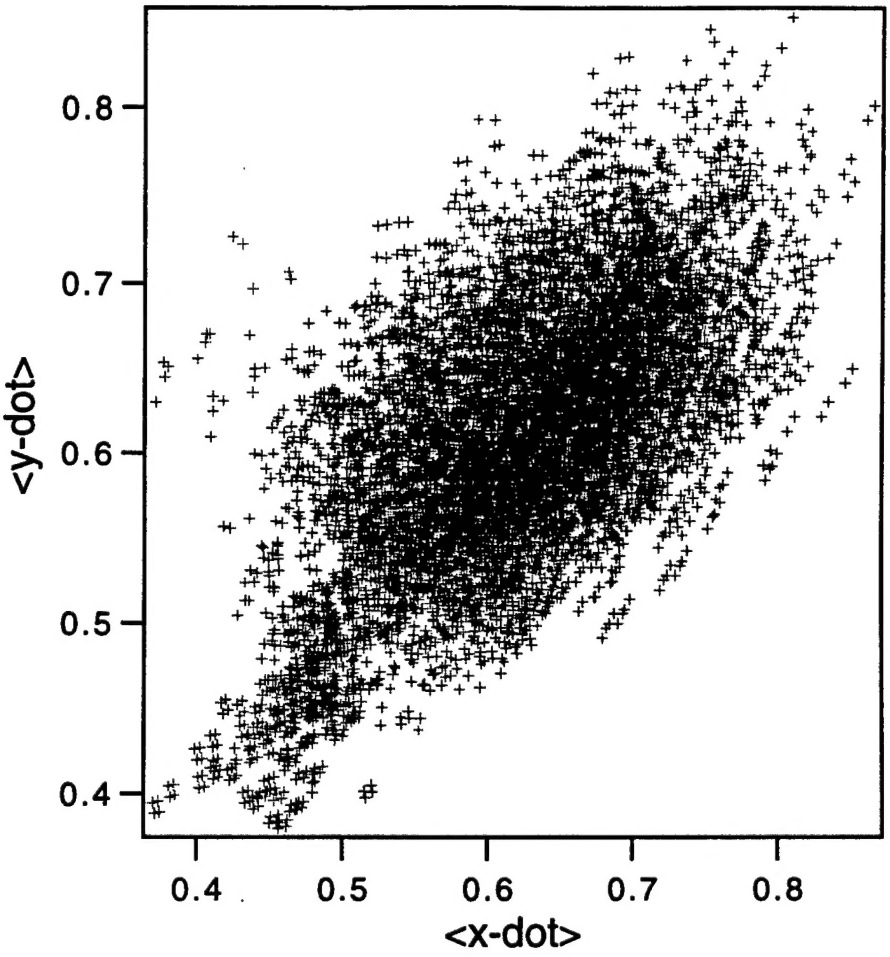

Fig. 3(b) — Corresponding x and y average dot products.

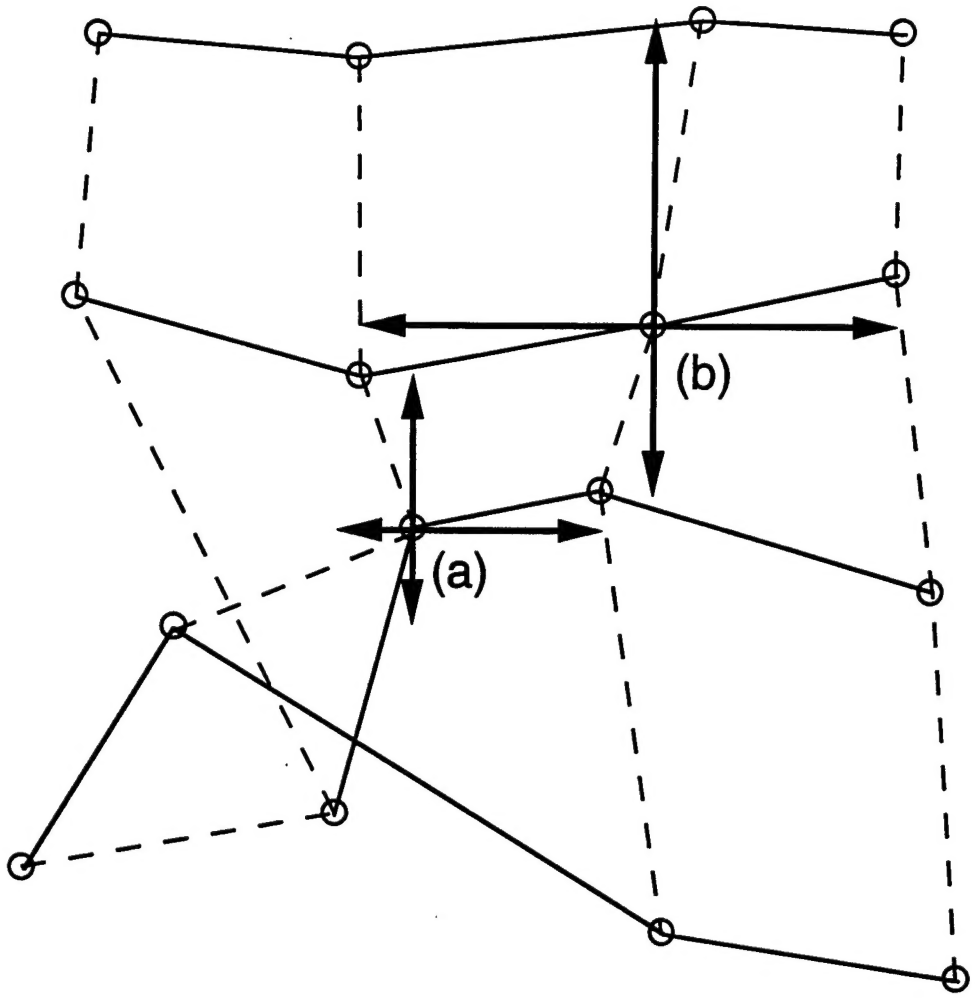

Fig. 4 — 4 × 4 MLG for 16 nodes. Solid and dashed lines represent x and y links respectively. Arrows indicate maximum displacements of selected nodes before monotonicity conditions are violated. Node (a) is at a relatively unstable region of the grid while node (b) is at a stable region.

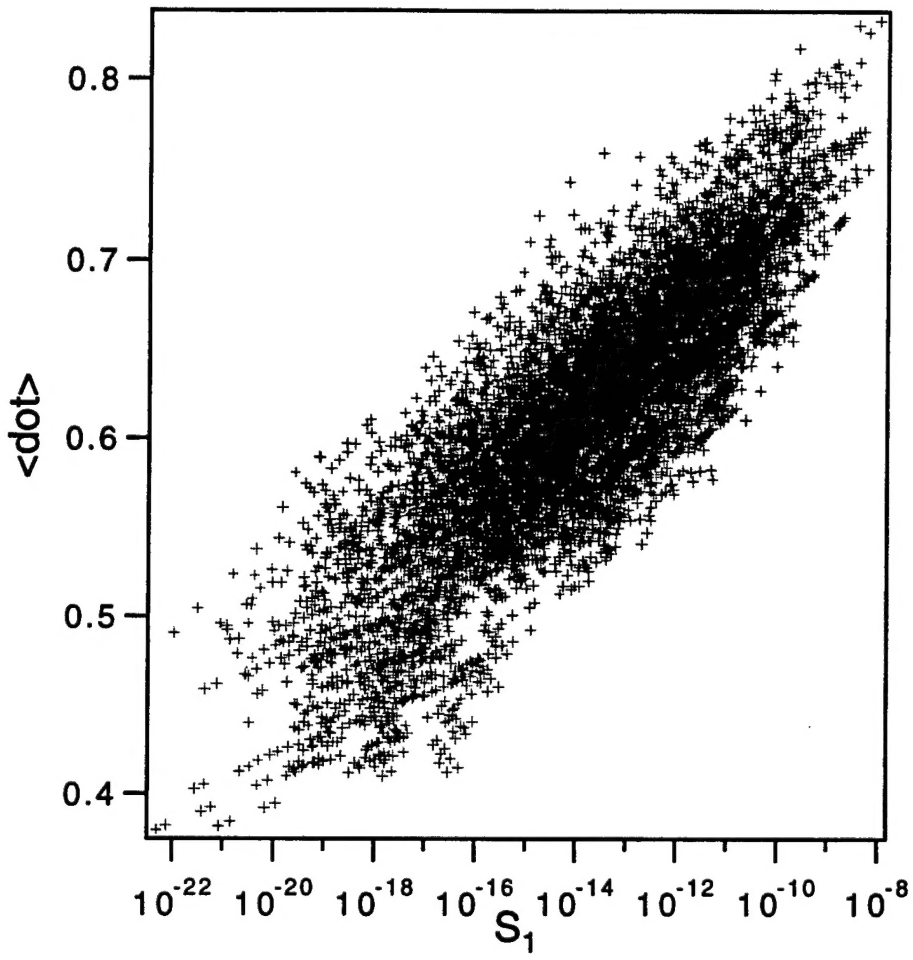

Fig. 5(a) — Average dot products plotted versus MLG stability for all allowed 5 × 5 MLG's for a set of 25 nodes. The same spatial configuration of nodes was used to generate data for this figure and Figure 3.

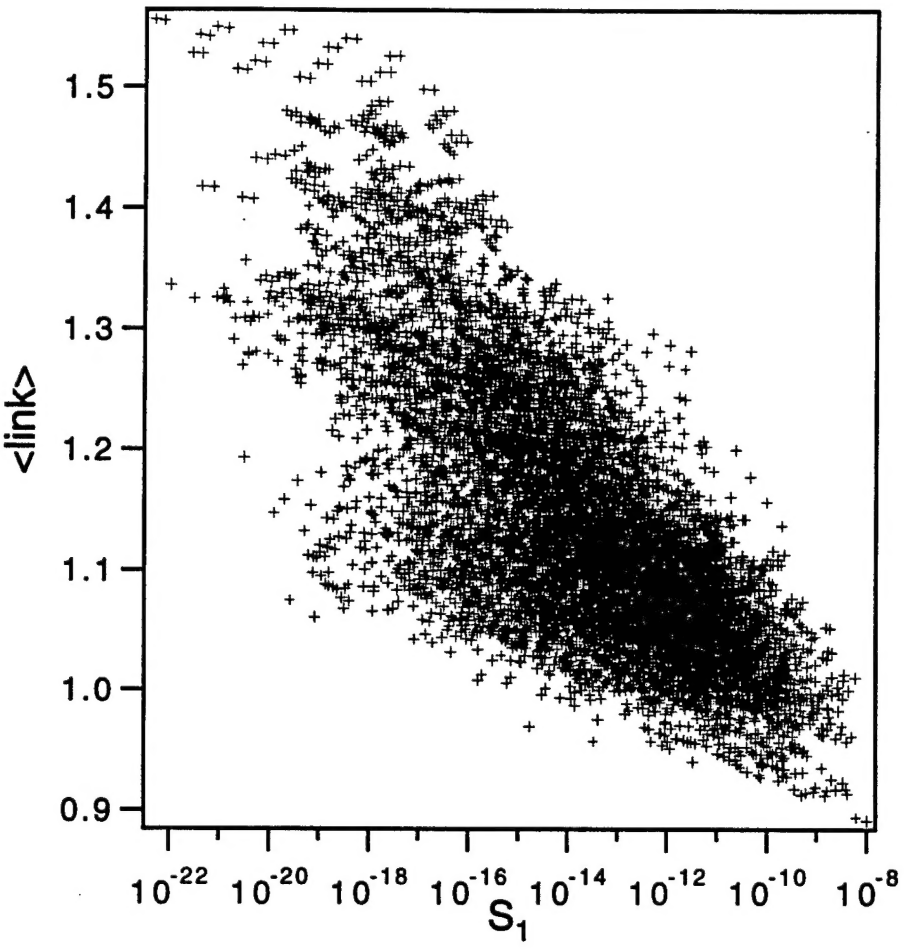

Fig. 5(b) — Average link lengths plotted versus MLG stability. The same spatial configuration of nodes was used to generate data for this figure and Figure 3.